

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-80813

(43)公開日 平成5年(1993)4月2日

(51)IntCl.⁵

G 0 5 B 19/05

G 0 6 F 9/06

識別記号

A 7361-3H

4 3 0 V 8944-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数4(全10頁)

(21)出願番号

特願平3-241085

(22)出願日

平成3年(1991)9月20日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71)出願人 000233217

日立京葉エンジニアリング株式会社

千葉県習志野市東習志野7丁目1番1号

(72)発明者 押賀 孝幸

千葉県習志野市東習志野7丁目1番1号

株式会社日立製作所習志野工場内

(72)発明者 加藤 富和

千葉県習志野市東習志野7丁目1番1号

日立京葉エンジニアリング 株式会社内

(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 プログラマブルコントローラのプログラミング装置

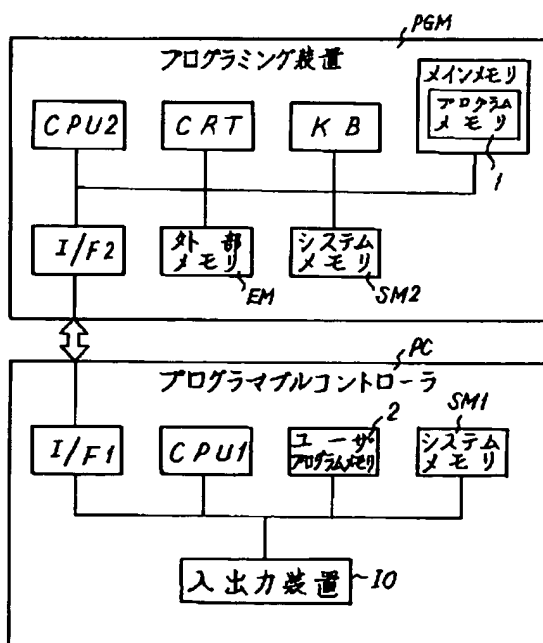
(57)【要約】

【目的】本発明は、プログラマブルコントローラとプログラミング装置内のメモリを同時編集し、両者のメモリ不一致をなくすことにある。また、プログラムをデバッグするときに、メモリ内容の転送処理時間を短縮することにある。

【構成】プログラマブルコントローラPCからプログラミング装置PGMに、ユーザプログラムメモリ2の内容を転送した後、従来行っていたオンラインの編集処理を、プログラミング装置PGM側についても追加させることにより、両者のメモリ1、2の記憶内容を同一とする。

【効果】メモリ転送作業の削減、メモリ不一致によるトラブル防止の効果がある。

(図 2)



1

【特許請求の範囲】

【請求項1】プログラマブルコントローラに設けたユーザプログラムメモリと対応するプログラムメモリをプログラミング装置内に備え、前記プログラミング装置により前記いずれか一方のメモリの内容を変更可能にしたものにおいて、

前記プログラミング装置に、前記ユーザプログラムの変更に応答し、前記両メモリの内容を同一内容に変更する記憶内容更新手段を具備してなるプログラマブルコントローラのプログラミング装置。

【請求項2】プログラマブルコントローラに設けたユーザプログラムメモリと対応するプログラムメモリをプログラミング装置内に備え、前記プログラミング装置により前記いずれか一方のメモリの内容を変更可能にしたものにおいて、

前記プログラミング装置に、当該プログラミング装置によりいずれか一方のメモリの内容を変更したことに応答し、他方のメモリの内容を同一内容に変更する記憶内容更新手段を具備してなるプログラマブルコントローラのプログラミング装置。

【請求項3】プログラマブルコントローラ内のユーザプログラムメモリにシーケンスプログラムを作成または編集するオンラインモードと、プログラミング装置本体のプログラムメモリにシーケンスプログラムを作成または編集するオフラインモードを有するものにおいて、前記オフラインモード、前記オンラインモード、及び前記両メモリを並列して編集するオンライン並列編集モードの各モードを選択可能に構成し、起動時に選択されたモードは、そのモードが終了するまで当該モードの機能を続行することを特徴とするプログラマブルコントローラのプログラミング装置。

【請求項4】プログラミング装置の起動時において、オンライン並列モードの選択に応答し、プログラマブルコントローラ内のメモリのシーケンスプログラムと、プログラミング装置本体のメモリのシーケンスプログラムの比較照合を実行し、両者が一致していれば前記オンライン平行編集モードを実行し、一致していなければ、操作者に対して警告表示、または一致操作を促す表示を行う記憶内容比較手段を具備して成ることを特徴とする請求項3記載のプログラマブルコントローラのプログラミング装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、プログラマブルコントローラのユーザプログラムメモリに記憶するシーケンスプログラムの作成、あるいはこれの編集を表示画面との対応で行うプログラマブルコントローラのプログラミング装置に関する。

【0002】

【従来の技術】一般的にプログラマブルコントローラ

2

は、シーケンスプログラムであるユーザプログラムを記憶するユーザプログラムメモリを備えている。そして、一般的に、ユーザプログラムメモリに記憶するシーケンスプログラムは、プログラミング装置によって作成、編集され記憶される。このため、プログラミング装置は、シーケンスコントローラ内に設けたユーザプログラムメモリと同様構成のプログラムメモリを備えている。

【0003】このプログラミング装置にてプログラマブルコントローラのユーザプログラムを作成、または編集する際には、まずプログラマブルコントローラのユーザプログラムメモリの内容をプログラミング装置内のプログラムメモリに転送する。そして、プログラミング装置でこのプログラムメモリの内容の編集等の作業終了後に、その編集後の内容を一括して再びプログラマブルコントローラのユーザプログラムメモリに転送する方法を取っている。

【0004】一方、プログラミング装置は、前記のように内蔵しているプログラムメモリの内容を指定してその編集を実行できることは勿論、プログラマブルコントローラのユーザメモリを指定してその編集を実行できるよう構成されている。一般的に、前者をオフラインモード、あるいはオフラインプログラミングと称し、後者をオンラインモード、あるいはオンラインプログラミングと称している。

【0005】この種の技術の参考となるものには、特開昭56-11502号を挙げることができる。

【0006】

【発明が解決しようとする課題】上記した従来のものは、プログラマブルコントローラのユーザプログラムメモリの内容と、プログラミング装置のプログラムメモリの内容とを同時編集するという点についての配慮がされておらず、両者のメモリを同一内容にするには、プログラマブルコントローラのユーザプログラムメモリの内容をプログラミング装置のプログラムメモリへ転送し、その編集後、再び元のユーザプログラムメモリへ転送するという2度の転送が必要であった。

【0007】このため、特にシーケンスプログラムのデバックを操作者が実施する場合、プログラムを変更する毎に、上記転送作業をしなければならず、余分な操作と転送処理の時間待ちを強いられていた。

【0008】さらに、オンラインモードによるプログラミングでは、プログラマブルコントローラ側のユーザプログラムメモリの内容が編集される為、その編集後にプログラマブルコントローラのユーザプログラムメモリの内容をプログラミング装置へ転送することを忘れると、両者のメモリ内容が一致なくなるという問題が生じていた。

【0009】本発明の目的は、編集作業終了後に一括してメモリ転送しなくても、両者のメモリを常に同一にすることが可能なプログラマブルコントローラのプログラ

50

3

ミング装置を得ることにある。そして、オンラインプログラミング終了後に生ずるメモリ不一致を無くすことにある。

【0010】本発明の他の目的は、プログラマブルコントローラのシーケンスプログラムのデバック作業を操作者が実施する場合に、メモリ内容の転送処理に要する時間を短縮可能なプログラムコントローラのプログラミング装置を得ることにある。

【0011】

【課題を解決するための手段】上記の目的を達成するため、本発明においては、プログラマブルコントローラに設けたユーザプログラムメモリと対応するプログラムメモリをプログラミング装置内に備え、前記プログラミング装置により前記いずれか一方のメモリの内容を変更可能にしたものにおいて、前記プログラミング装置に、前記ユーザプログラムの変更に応答し、前記両メモリの内容を同一内容に変更する記憶内容変更手段を具備したことを特徴とする。

【0012】本発明の好適な実施例によれば、最初にプログラマブルコントローラ内のユーザプログラムメモリを一括してプログラミング装置のユーザプログラムメモリへ転送の後、オンラインプログラミングで行っていた1回路内のステップ単位の転送処理を、プログラミング装置内のユーザプログラムメモリについても行う構成としたことを特徴とする。

【0013】更に他の好適な実施例によれば、最初にプログラミング装置のユーザプログラムメモリを一括してプログラマブルコントローラ内のユーザプログラムメモリへ転送の後、オンラインプログラミングで行っていた1回路内のステップ単位の転送処理を、プログラミング装置内のユーザプログラムメモリについても行う構成としたことを特徴とする。

【0014】

【作用】上記のようにすれば、編集した内容が、直ちにプログラマブルコントローラ内のユーザプログラムメモリとプログラミング装置内のプログラムメモリに反映される為、両者のメモリ不一致によるトラブルの発生を未然に防止することができる。

【0015】また、編集前のプログラマブルコントローラ内のメモリのシーケンスプログラムと、プログラミング装置本体のメモリのシーケンスプログラムを一致させる作業が初期処理において完了しているので、編集中は、変更した部分のメモリの内容だけを転送すればよく、転送処理に要する時間を短縮できる。

【0016】

【実施例】図1は本実施例におけるプログラミング装置の初期画面の状態を示す図である。プログラミング装置でプログラミングソフトを立ち上げると、モード機能を選択する図1(a)の初期画面を表示する。ここにおいて、選択番号「1」はオフラインモード、すなわちプロ

4

グラミング装置に内蔵したプログラムメモリの内容を指定してその編集を実行するモードである。選択番号

「2」はオンラインモード、すなわちプログラマブルコントローラのユーザプログラムメモリを指定してその編集を実行するモードである。選択番号「3」は本実施例によって新たに追加したモードであり、プログラマブルコントローラ内のユーザプログラムメモリとプログラミング装置のプログラムメモリとを指定して、これら両メモリを並列して編集を実行するモードであり、表示画面上においては新オンラインモードと称してある。プログラミング装置は、図1に示す初期画面の中で、新オンラインモード「3」が選択されると、プログラマブルコントローラ内のユーザプログラムメモリに記憶されたシーケンスプログラムと、プログラミング装置本体のプログラムメモリのシーケンスプログラムが一致しているか否かを確認し、一致していれば図1(c)の画面を表示して新オンラインモードによるプログラミング処理を開始する。もし、一致していなければ図1(b)の画面を表示して操作者に対して、上記一致のためのメモリ転送方向を問い合わせる。

【0017】図1(b)において、選択番号「1」が選択されると、プログラマブルコントローラ内のユーザプログラムがプログラミング装置へ転送され、プログラマブルコントローラ内のユーザプログラムメモリの記憶内容と、プログラミング装置内のプログラムメモリの記憶内容との記憶内容が一致し、その後、図1(c)の画面表示となる。また、図1(b)において、選択番号「2」が選択されると、プログラミング装置内のプログラムメモリの内容、すなわちシーケンスプログラムがプログラマブルコントローラへ転送され、両者の内容が一致し、その後、図1(c)の画面表示となる。

【0018】図2は本発明の一実施例を示す装置の全体の構成図である。この図において、PGMはプログラミング装置を示し、PCはプログラマブルコントローラの構成を示す。プログラミング装置PGMはプログラミングのための制御プログラムを記憶するシステムメモリ、シーケンスプログラムおよびデータ等を記憶する内部記憶装置としてのメインメモリMM、入力装置としてのキーボードKB、表示装置CRT、外部記憶装置としての外部記憶メモリEM、作成したシーケンスプログラムを送信、あるいは受信するシリアルインタフェースIF2、およびプログラミング装置PGM全体の制御を行う中央処理部CPU2を備えている。プログラマブルコントローラPCは、プログラミング装置PGMから送信されたシーケンスプログラムを受信、あるいはプログラミング装置PGMへシーケンスプログラムを送信するインタフェースIF1、この受信したシーケンスプログラムを格納するユーザプログラムメモリ2、シーケンス制御のための制御プログラムを記憶するシステムメモリSM1、このシステムメモリSM1の内容に基づきユーザア

5

ログラムメモリ2に格納されたユーザプログラムを実行する中央処理部CPU1、および制御対象との間でデータを授受する入出力部IOを備えている。

【0019】プログラミング装置PGM内のメインメモリMMには、ユーザプログラムを構成するためユーザプログラムメモリ2と対応したプログラムメモリを割り当てる。また、ユーザプログラムの編集等のため編集用のエリアを割り当てる。

【0020】図1において、新オンラインプログラミングを選択すると、プログラマブルコントローラPC内にあるユーザプログラムメモリ2からインタフェースIF2を介してプログラミング装置PGM内のプログラムメモリ1にその内容が転送される。

【0021】次に、オンライン処理の起動を開始し、ユーザプログラムメモリ2における編集回路をメインメモリMMのプログラムメモリ1以外の編集用メモリエリアに転送し、回路変更作業を実行する。この回路変更作業で変更されたステップの内容をユーザプログラムメモリ2に転送する。そして、同じ転送内容をプログラムメモリ1についても実行し、1回路内のステップ変更作業が終了する。同回路内において他のステップを変更するのであれば、この一連の作業を繰り返すことになる。

【0022】図3は本実施例により新たに追加した処理を示すフローチャートであり、これは図1(a)において項目番号3が選択されてことによって起動される。このフローチャートにおいて、ステップ3a、3b、3c、3dは記憶内容比較手段を構成し、ステップ3e、3f、…、3kは記憶内容更新手段を構成する。この処理は起動されることによって、まず、ステップ3aでは、プログラマブルコントローラPCのユーザプログラムメモリ2の記憶内容と、プログラミング装置PGM内のプログラムメモリ1との一致状況を検知する。ここで、仮に両者のメモリ内容が一致していなければ、ステップ3bで、表示画面上に両メモリが不一致である旨の表示を行う。そして、ユーザに対し、両メモリを一致させるためのデータの転送方向を選択させる画面を表示、すなわち図1(b)に示した画面を表示する。ここで、ユーザが「1」を選択したとすると、ステップ3cでプログラマブルコントローラPCのユーザプログラムメモリ2からプログラミング装置PGMのプログラムメモリ1へその記憶内容を転送記憶する。これにより、プログラミング装置PGMのプログラムメモリ1の記憶内容は、プログラマブルコントローラPCのユーザプログラムメモリ2の内容と一致することとなる。また、ステップ3dでユーザが「2」を選択したとすると、ステップ3dでプログラミング装置PGMのプログラムメモリ1からプログラマブルコントローラPCのユーザプログラムメモリ2へその記憶内容を転送記憶する。これにより、プログラミング装置PGMのプログラムメモリ1の記憶内容は、プログラマブルコントローラPCのユーザプログラ

6

ムメモリ2の内容と一致することとなる。ステップ3c、あるいはステップ3dの実行後、処理をステップ3eへ移す。ステップ3aにおいて両メモリ1、2の内容が一致していれば、ステップ3b、3c、3dを実行することなく、処理をステップ3eへ移す。

【0023】ステップ3eにおいては、本実施例によるオンラインモードの開始処理を実行する。周知のように、シーケンスプログラムは一般的には1回路単位でその編集作業を実行する。この実施例も1回路単位で編集する場合について示してある。ステップ3fでは、編集対象となる回路番号等の入力をユーザに要求し、プログラミング装置PGMからのユーザ入力にตอบสนองして、対象となる1回路をプログラミング装置PGMのメインメモリ4に予め割り付けた記憶エリア内に記憶格納する。続く、ステップ3gにおいては、この記憶エリア内の回路をユーザの編集操作に基づいて回路中の1ステップ分の記憶内容を編集する。これが終了すると、ステップ3hで、この編集内容をプログラマブルコントローラPCのユーザプログラムメモリ2に転送し、続いてステップ3iでは同一の編集内容をプログラミング装置PGMのプログラムメモリ1へ転送する。ステップ3jでは、当該回路の編集作業が終了したか否かを判定し、終了していなければ処理をステップ3gへ移し、上記の処理を繰り返す。終了するのであればステップ3kへ処理を移し、編集作業を終了するか否かを判定し、終了していなければ処理をステップ3fへ移し、上記の処理を繰り返す。そして、終了するのであれば、以上の編集作業は終了する。

【0024】以上の実施例におけるオンラインプログラミングの内部処理の詳細を図4～図6に模式図で示す。図4は回路編集後のステップ数が増加した場合の処理を示したものであり、図4(a)は、編集後のデータ処理前の状態を示し、プログラマブルコントローラPC内のユーザプログラムメモリ10から1回路分のデータがプログラミング装置PGM内のメモリ9に転送され、変更された1回路分のステップがステップ数7であり、それがプログラミング装置9の中にあることを示す。斜線で示した部分が編集後に増加したステップ数11である。

【0025】図4(b)は、増加分のステップ転送処理を示し、増加したステップ部分だけをユーザプログラムメモリ10とプログラミング装置PGM内のメモリ8に転送する処理を示す。この場合、転送する前にあらかじめ増加分のステップ数だけずらして転送処理を行う。一般のオンラインプログラミングに対し、転送処理15が追加されている。

【0026】図4(c)は、1回路の残りステップの転送処理を示したものである。1回路の残されたステップ12をユーザプログラムメモリ10とプログラミング装置PGM内のメモリ8へ転送する。一般のオンラインプログラミングに対し、転送処理15が追加されている。図

4の処理の流れは、図4(a)、図4(b)、図4(c)の順に行われる。

【0027】図5は、回路編集後のステップ数が同一の場合の処理を示したものであり、図5(a)は、編集後のデータ処理前の状態を示したものである。編集したステップ13が実質転送ステップになる。図5(b)は、転送ステップ先頭アドレスの検出処理を示しており、プログラマブルコントローラPGM内の変更前回路メモリ14とメモリ7とを照合し、照合不一致アドレスをADR'とする。ここが転送開始アドレスになる。図5(c)は、転送ステップ最終アドレスの検出処理を示しており、回路の最終ステップから一致しているステップ数16を求める。実質転送ステップ13は、 $7-16-(ADR'-ADR)$ になる。図5(d)は、ステップ転送処理を示しており、オンラインプログラミングではなかった13を8に転送する処理に16を追加したものである。図5の処理の流れは、図5(a)、図5(b)、図5(c)、図5(d)の順に行われる。

【0028】図6は、回路変更後のステップ数が減少した場合の処理を示したものであり、図6(a)は、編集後のデータ処理前の状態を示したものである。15が減少したステップ数になる。図6(b)は、減少ステップ数分のプログラムの削除を示したもので、オンラインプログラムになかった15を削除処理を追加した処理になっている。図6(c)は、変更回路の転送を示したものであり、図6(b)によって減少された10と8のメモリに変更回路16を転送する。オンラインプログラミングにはなかった15を追加した処理である。図6の処理の流れは、図6(a)、図6(b)、図6(c)の順に行われる。

【0029】以上、実施例のようにすれば、編集した内容が、自動的に直ちにプログラマブルコントローラPC内のユーザプログラムメモリ2とプログラミング装置PGM内のプログラムメモリ1に反映される為、両者のメモリ不一致によるトラブルの発生を未然に防止することができる。

【0030】また、編集前のプログラマブルコントローラPC内のメモリのシーケンスプログラムと、プログラミング装置PGM本体のメモリのシーケンスプログラム

を一致させる作業が初期処理において完了しているので、編集中は、変更した部分のメモリの内容だけを転送すればよく、転送処理に要する時間を短縮できる。

【0031】

【発明の効果】以上の説明から明らかなように、本発明によれば、オンラインとオフラインの機能を兼ね備えた機能により、メモリ転送の操作を省くことができ、プログラムのデバッグに要する時間を短縮できる。

【0032】また、メモリ不一致によるトラブルを防ぐことができる。

【図面の簡単な説明】

【図1】本発明の一実施例におけるプログラミング装置の表示画面の表示例を示す図である。。

【図2】本発明の一実施例を示すプログラミング装置とプログラマブルコントローラの構成図を示すブロック図である。。

【図3】本発明の一実施例を示す処理のフローチャートである。

【図4】回路変更後のステップ数が増加した場合の処理を示した模式図である。

【図5】回路変更後のステップ数が同一の場合の処理を示した模式図である。

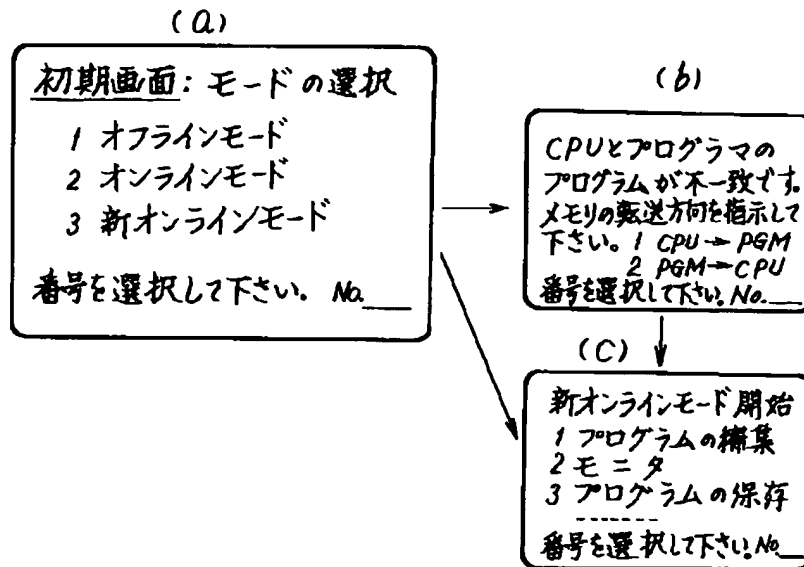
【図6】回路変更後のステップ数が減少した場合の処理を示した模式図である。

【符号の説明】

1：プログラマブルコントローラ内のユーザプログラムメモリ、2：プログラミング装置内のプログラムメモリ、IF1、IF2：インタフェース、MM：プログラマブルコントローラ内のメインメモリ、7：1回路分のステップ数、8：プログラミング装置内の編集前のユーザプログラムメモリ、9：プログラミング装置内のユーザプログラムメモリ、10：プログラマブルコントローラ内の編集前のユーザプログラムメモリ、11：編集後に増加したステップ数、12：増加分以外の編集後の回路、13：編集後のステップ、14：プログラマブルコントローラ内の編集前のユーザプログラムメモリ、15：転送処理、16：編集後のプログラム、3a、3b、3c、3d：記憶内容比較手段、3e、3f、…、3k：記憶内容更新手段

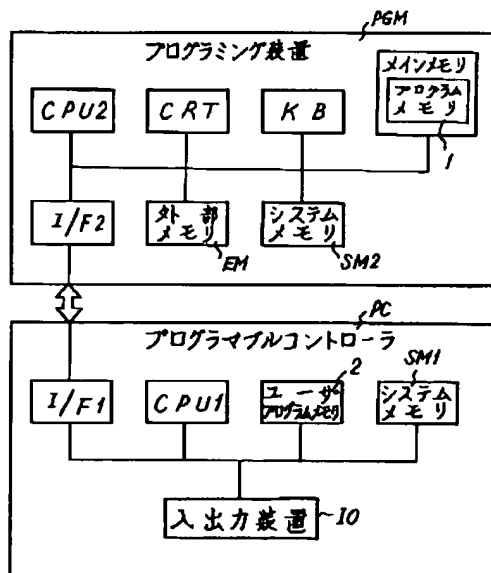
【図1】

(図 1)



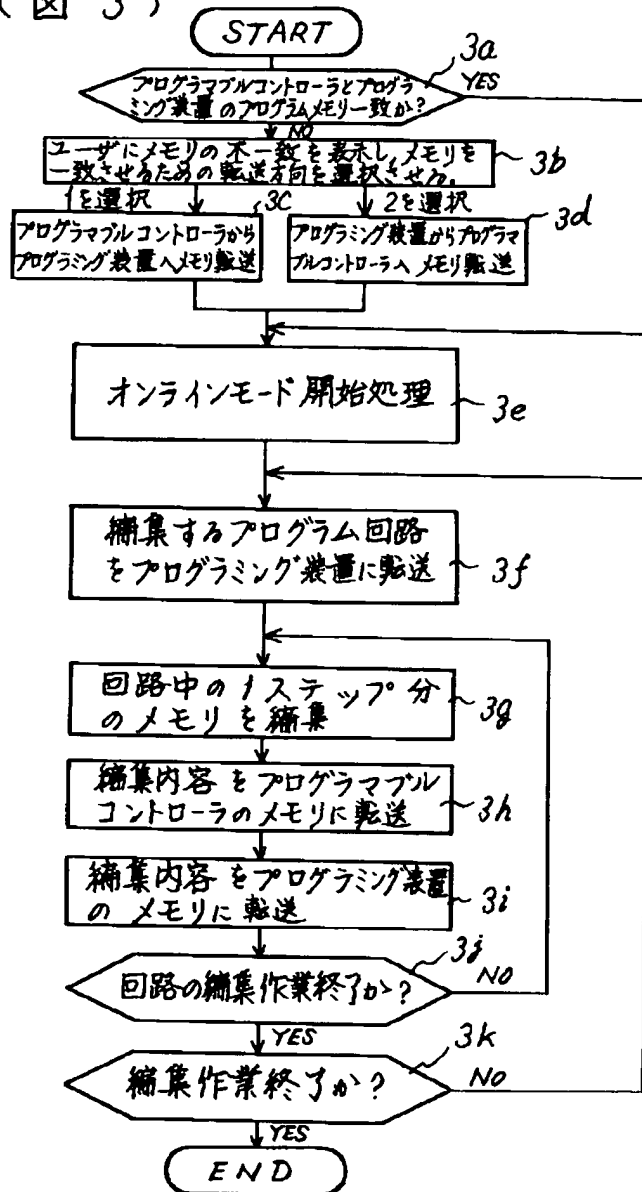
【図2】

(図 2)



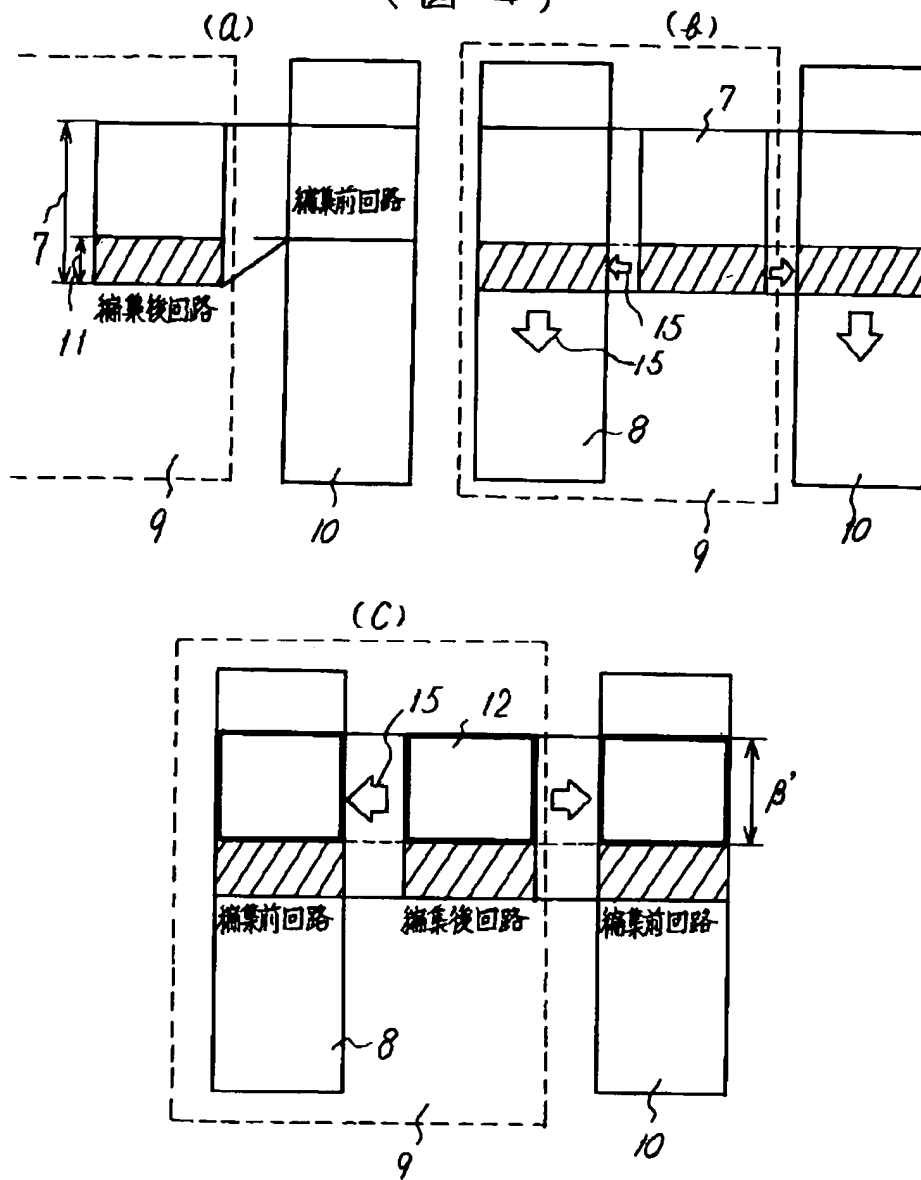
【図3】

(図 3)

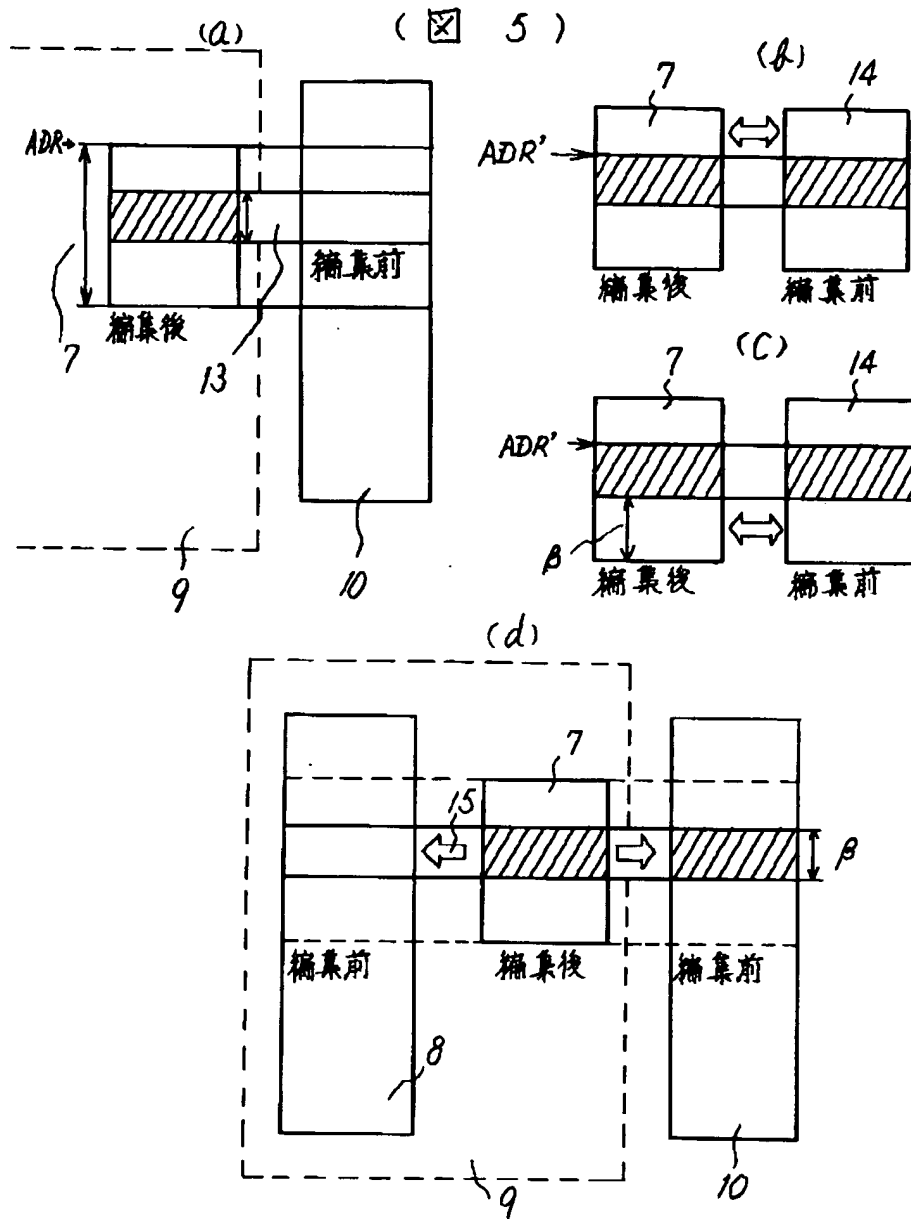


【図4】

(図 4)



【図5】



【図6】

(図 6)

